

## Para saber más: control de acceso a anotaciones

Otra forma de restringir el acceso a ciertas funciones, según el perfil del usuario, es usar una función de Spring Security conocida como Method Security, que funciona con el uso de anotaciones en los métodos:

```
@GetMapping("/{id}")
@Secured("ROLE_ADMIN")
public ResponseEntity detallar(@PathVariable Long id) {
    var medico = repository.getReferenceById(id);
    return ResponseEntity.ok(new DatosDetalladoMedico(medico));
}
```

[COPIA EL CÓDIGO](#)

En el ejemplo de código anterior, el método se anotó con `@Secured("ROLE_ADMIN")`, de modo que sólo los usuarios con el rol ADMIN pueden activar solicitudes para detallar a un médico. La anotación `@Secured` se puede agregar en métodos individuales o incluso en la clase, lo que sería el equivalente a agregarla en todos los métodos.

¡Atención! Por defecto esta característica está deshabilitada en Spring Security, y para usarla debemos agregar la siguiente anotación en la clase `Securityconfigurations` del proyecto:

```
@EnableMethodSecurity(securedEnabled = true)
```

[COPIA EL CÓDIGO](#)

Puede obtener más detalles sobre la función de seguridad del método en la documentación de Spring Security, disponible en:

- [Method Security \(https://docs.spring.io/spring-security/reference/servlet/authorization/method-security.html\)](https://docs.spring.io/spring-security/reference/servlet/authorization/method-security.html).